

谢金宝 编
张 勇

C语言程序设计和应用

(附自学软件包)



上海交通大学出版社

C 语言程序设计和应用

(附自学软件包)

谢金宝 张勇 编著

上海交通大学出版社

(沪)新登字 205 号

内 容 简 介

本书从读者的角度出发,由浅入深地围绕着解答这样一个最基本的问题:如何利用 C 语言把思想变为程序?或者要将思想变为程序,利用 C 语言怎么做?

本书力图做到通俗易懂,且强调可操作性。遵循“问题—理论—实践”的基本框架,目的是为了能使读者从一开始就能模仿着编写出各种符合想法的应用程序。

全书共十二章,分基本篇和应用篇,各章节均有一定数量的程序实例,结合程序提出概念,并逐行解释语句的含义。每篇附有相应的软件包,凡是书中出现的程序实例(包括习题解答在内)均收录在软件包中,既便于课堂教学,也便于广大读者自学。

C 语言程序设计和应用

出版:上海交通大学出版社

(上海市华山路 1954 号 邮政编码:200030)

发行:新华书店上海发行所

印刷:上海交通大学印刷厂

开本:787×1092(毫米)1/16

印张:18.75 字数:465000

版次:1995 年 7 月 第 1 版

印次:1995 年 8 月 第 1 次

印数:1—5000

ISBN 7-313-01477-5/TP·271 定价:26.00 元(附软盘)

前 言

随着 UNIX 和 XNIEX 操作系统的广泛使用，作为这两种操作系统的主语言：C 语言，正在越来越受到人们的青睐。C 语言不仅适合编写系统程序，而且也适合编写应用程序。C 语言简洁、灵活、效率高、移植性好。正是由于这些特点，无论初涉计算机领域还是从事计算机软件工作多年的人均对它发生了浓厚的兴趣。许多人想学，但一翻开书，满眼的概念和理论，往往使他们望书却步。

为了使更多的人尽快熟悉 C 语言，并且掌握和利用 C 语言来进行程序设计(包括编写应用程序和系统程序)，本书将从读者的角度出发，由浅入深地围绕着解答这样一个最基本的问题：如何利用 C 语言把思想变为程序？或者要将思想变为程序，利用 C 语言怎么做？本书力图做到通俗易懂，且强调可操作性。遵循“问题—理论—实践”的基本框架，目的是为了使得读者从一开始就能模仿着编写出各种符合想法的应用程序。全书分基本篇和应用篇，各章节均有一定数量的程序实例，结合程序提出概念，并逐行解释语句的含义。每篇附有相应的软件包，凡是书中出现的程序实例(包括习题解答在内)均收录在软件包中，既便于课堂教学，也便于广大读者自学，因为读者可直接使用软件包中某些符合需要的程序，或根据实际用途对软件包中部分程序作少量修改而省时省力地编写出一个新的应用程序，甚至十个几十个满意的程序。相信读者在使用本书及软件包的过程中，会逐步体会到事半功倍的效果，而这也正是编者的愿望，读者的需要。

本书共有十二章，第一至第十章由谢金宝编写，第十一至第十二章由张勇编写，习题部分由章桓明收集和整理。

本书的编写是一种新的尝试，软件包中的程序虽然涉及面颇广，但也难免挂一漏万。还望得到读者及专家的建议和指正。

编者
1995.2

软件包使用方法

本书附有一软件包，提供在本书中出现的所有例题和习题答案。每个例题和习题对应一个文件名，例如 exam1__1.c 为例 1.1 的文件名，exam4__8.c 为例 4.8 的文件名；exs1__1.c 为习题 1.1 的习题答案文件名，exs4__8.c 为习题 4.8 的习题答案文件名。

软件包使用方法如下：

1. 将软件包中的文件复制到 c:\。
2. 键入 TC 命令，进入 TURBO C 主屏幕。
3. 打开 File 菜单，选择 Load 命令，此时会出现一文件名对话框，输入一个文件名，例如 C:\exam4__1.c 表示第四章第一个例题。
4. 按 F9 对源程序进行编译和连接。
5. 打开 Run 菜单，选择 Run 命令，执行程序。
6. 按 Alt+F5 观察程序运行结果。
7. 按任意键返回主屏幕。
8. 打开 File 菜单，选择 Quit 命令退出 TURBO C。

目 录

第一章 概述	1
§ 1.1 C 语言的特点	1
§ 1.2 如何编写 C 语言程序	1
§ 1.3 如何建立源程序文件	2
§ 1.4 编译和运行 C 语言程序	5
§ 1.5 如何在屏幕上显示信息	8
习题.....	9
第二章 基本数据类型、运算符和表达式	11
§ 2.1 变量和数据类型	11
§ 2.2 常数	15
§ 2.3 运算符和表达式	19
§ 2.4 数据类型转换	30
§ 2.5 强制数据类型转换	32
习题.....	33
第三章 分支语句和循环语句	36
§ 3.1 条件分支语句	36
§ 3.2 switch 和 break 语句	42
§ 3.3 循环语句	44
§ 3.4 转移语句和空语句	52
习题.....	55
第四章 数组	57
§ 4.1 数组变量	57
§ 4.2 整型数组	57
§ 4.3 字符数组	62
§ 4.4 多维数组	64
习题.....	70
第五章 函数	72
§ 5.1 库函数	72
§ 5.2 函数基础	77
§ 5.3 非整型函数	81

§ 5.4 参数传递方式	84
§ 5.5 变量的存储类和作用域	88
§ 5.6 不同文件中的程序的编译和运行	97
§ 5.7 递归函数	98
§ 5.8 分程序结构	101
§ 5.9 预处理程序	102
习题	107
第六章 指针	110
§ 6.1 指针和地址	110
§ 6.2 指针和函数参数	113
§ 6.3 指针和数组	114
§ 6.4 指针运算	117
§ 6.5 指针和字符串	118
§ 6.6 指针数组	120
§ 6.7 函数指针	124
§ 6.8 命令行参数	127
习题	131
第七章 结构	133
§ 7.1 结构变量	133
§ 7.2 结构与函数	137
§ 7.3 结构与数组	140
§ 7.4 指向结构的指针	148
§ 7.5 引用自身的结构	150
§ 7.6 位段	164
§ 7.7 类型定义	166
§ 7.8 联合	167
§ 7.9 枚举型	168
习题	170
第八章 输入和输出	171
§ 8.1 字符输入和输出	171
§ 8.2 格式输出和输入	172
§ 8.3 文件输入和输出	177
习题	183
第九章 UNIX 文件系统	185
§ 9.1 概述	185

§ 9.2 文件的打开和关闭、建立和撤消	186
§ 9.3 文件的读和写	187
§ 9.4 文件的随机访问	189
§ 9.5 UNIX 文件系统	190
§ 9.6 文件结构和目录文件	192
习题	196
第十章 DOS 文件系统	198
§ 10.1 低层输入 / 输出(I / O)	198
§ 10.2 打开文件	198
§ 10.3 读写文件	200
§ 10.4 关闭和撤消文件	200
§ 10.5 DOS 文件系统	201
第十一章 C 语言高级应用程序的开发基础	206
§ 11.1 概论	206
§ 11.2 C 语言的标准库函数	206
§ 11.3 C 语言访问 DOS 开发环境下的各种系统资源	227
§ 11.4 C 语言在 DOS 开发环境下屏幕字符显示的操作	236
§ 11.5 C 语言在 DOS 开发环境下对图形的操作	241
§ 11.6 DOS 环境下的 C 语言编程开发程序软件包 Turbo C	250
第十二章 C 语言高级应用程序开发	259
§ 12.1 C 语言在汉字显示技术中的应用	259
§ 12.2 C 语言在计算机音乐技术中的应用	261
§ 12.3 C 语言在计算机显示菜单设计中的应用	262
§ 12.4 C 语言在计算机通讯技术中的应用	266
§ 12.5 C 语言在开发工具软件中的应用	271
附录 A ASCII 码集	273
附录 B C 语言汇总	275
附录 C IBM 字符代码表	288
参考文献	292

第一章 概述

§ 1.1 C 语言的特点

D.M.Ritchie 在 1971 年编写了第一个 C 语言编译程序, 1972 年 C 语言便投入了使用。1973 年 K.Thompson 和 D.M.Ritchie 用 C 语言编写 UNIX 操作系统, 奠定了 UNIX 操作系统的基础。今天 C 语言不仅作为 UNIX 操作系统的主语言, 而且已成为广泛使用的程序设计语言。

C 语言受到越来越多的人的青睐是由于 C 语言的特点所决定的。那末, C 语言有哪些特点呢?

C 语言的第一个特点是它的模块结构。用 C 语言编写的程序由若干个函数组成, 每一个函数完成一个或多个功能, 该函数称为模块。这种模块结构的程序设计不仅程序结构清晰、易读、而且容易调试和裁剪。

C 语言的第二个特点是它的紧凑性。与其它高级程序设计语言相比, C 语言显得紧凑简洁。例如: 欲使某个变量的值增 1, 在 FORTRAN 语言中表达为 $I=I+1$, 在 PASCAL 语言中表达为 $I:=I+1$, 而在 C 语言中表达为 $i++$ 。

C 语言的第三个特点是它的效率高。C 语言提供了某些接近汇编语言的功能, 如地址处理、二进制数位运算以及指定使用寄存器存放变量等, 这些功能提高了语言生成代码的质量, 所以 C 语言适合用于编写系统程序和各类软件开发工具。

C 语言的第四个特点是它的可移植性。用 C 语言编写的程序具有良好的可移植性。所谓可移植性是指程序从一种类型的计算机移植到另一种类型的计算机。UNIX 操作系统和 C 编译程序在大、中、小和微型计算机上的广泛使用, 充分证明了 C 语言编写的程序具有良好的可移植性。

§ 1.2 如何编写 C 语言程序

下面的 C 语言程序非常简单, 它是在屏幕上显示“Hello everybody!”字符串。程序中的行号是另外加的, 只是为了说明的方便。

例 1.1

```
1:  /* exam1_1.c  A example */  
2:  main( )  
3:  {  
4:    printf("Hello everybody!");  
5:  }
```

对上面的程序说明如下:

1: 这是一个注释语句。在 C 语言中注释语句以“/*”开始, 以“*/”结束。注释语

句可以分行书写,但“/ *”和“* /”必须配对。例如第 1 行也可以写成:

```
/* simple  
example */
```

2: 第 2 行到第 5 行定义了一个名为 main 的函数。第 2 行定义函数名,函数名后面的一对括号不能省略。

3: 第 3 行的左花括号“{”定义函数正文的开始。

4: 第 4 行定义函数正文,又称函数体,函数正文描述要做的事情。该例中的函数正文是一个 printf 函数调用,printf 是一个库函数,它在屏幕上显示信息,本例显示双引号对中的字符串。

5: 第 5 行的右花括号“}”定义函数 main 的结束。该右花括号是与第 3 行的左花括号配对的。注意方向不要搞错。

例 1.1 表明 C 语言的函数具有下列的一般格式:

函数名(...)

...

{

函数体

}

其中...表示省略成分,可以出现也可以不出现。

函数体由若干个语句组成,每个语句以分号“;”结束。语句可以是一个函数调用,如例中的 printf 就是一个函数调用。C 语言提供了许多库函数,您可以按需要调用,不必自己编写程序。

按照习惯,C 语言的变量名,函数名用小写字母书写。

§ 1.3 如何建立源程序文件

文件是记录成字符流的集合,如果一个文件包含的内容是用 C 语言编写的程序,我们称这个文件为源程序文件。如何建立一个源程序文件呢?计算机软件环境不同,建立文件的方法也不同。考虑到 DOS 和 UNIX 操作系统的普遍使用,本书将以这两种环境为基础叙述程序的建立、编译和执行。

1.3.1 在 DOS TURBO C 环境下建立程序

按照下例步骤建立源程序文件 exam1_1.c:

1.假设 TURBO C 在子目录 TC 下面,则设置当前目录为 TC

C:\>cd tc

2.键入 tc exam1_1.c 或 exam1_1

c:\Tc>tc exam1_1c

若键入 exam1_1,则 TURBO C 会自动添加后缀.C,源程序文件均应以.C 作为文件后缀。键入上述命令后,便进入主屏幕。

主屏幕分四个窗口:主菜单窗口,编辑窗口,信息窗口和快速参考行窗口。主屏幕如

图 1.1 所示:

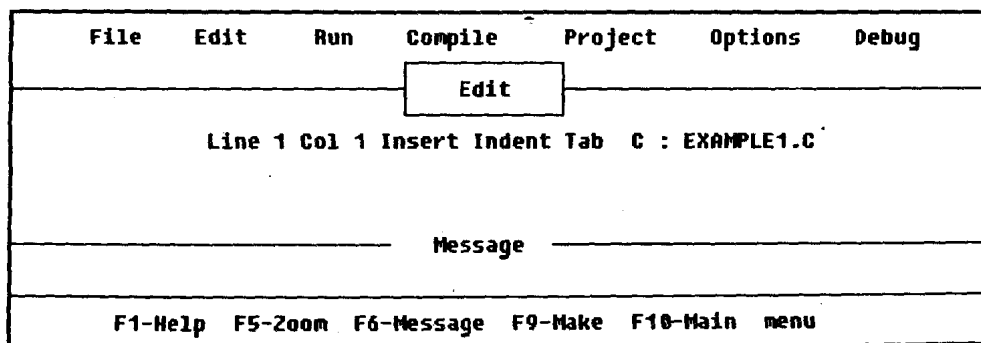


图 1.1 Turbo C 的主屏幕

进入主屏幕时, 光标定位在编辑窗口的第 1 行第 1 列上, 此时自动处于编辑状态, 您可以逐行输入程序。程序输入后, 可按功能键 F2 保存。此时并不退出编辑状态, 若要退出编辑状态, 则按功能键 F10, 光标定位在主菜单窗口。您可按←和→键移动光标到所需菜单, 例如移到 File 处, 按 Enter 键, 则在屏幕左上角会出现一个下拉窗口, 如下图所示:

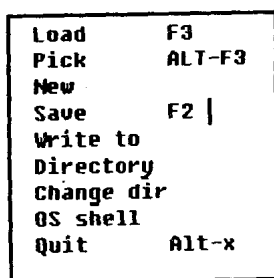


图 1.2 File 下拉窗口

按↓键, 直到光标定位在最后一行, 然后按 Enter 键, 退出编辑状态, 回到 DOS 状态。

如果在编辑中发现输入错误, 或漏输入一行, 或出现无用行, 则可再次进入主屏幕。一旦进入主屏幕后, 按 F10 键进入主菜单, 移动光标至 Edit 菜单, 按 Enter 键进入编辑状态。在编辑状态下, 按 F1 键可获得有关各种编辑子命令的求助信息。常用的编辑子命令如下:

插入字符——按 Insert 键, 进入插入方式。在光标所在位置前插入字符。再按 Insert 键结束插入方式。

插入一行——将光标移到某一行的第一列，按 Ctrl+N(即同时按 Ctrl 键和 N 键)，则在该行前出现一个空行，您可在空行处输入程序。若把光标移到某一行末端字符的右边，按 Ctrl+N 键，则在该行下面插入一行空行。

删除字符——按 BackSpace 键，删除当前光标位置前的一个字符，连续按以此键，则可连续删除字符。

删除一行——按 Ctrl+Y 键，删除光标所在行。

删除一行中的部分内容——按 Ctrl+Q+Y 键，删除自光标位置起至行末的所有字符。

窗口移动——按 PgDn 键，窗口向下移动一个屏幕。

按 PgUp 键，窗口向上移动一个屏幕。

欲获得关于编辑命令的详细说明，您可以阅读求助信息。

退出 Help 可按 ESC 键。

修改完毕后，按 F2 重写该源程序文件并存盘。

除按上面叙述的进入菜单的方法外，尚可使用 Alt+菜单首字母直接进入菜单的下拉窗口。下面是一些常用键的功能：

Alt+C 进入编译菜单

Alt+D 进入排错菜单

Alt+E 进入编辑菜单

Alt+F 进入文件菜单

Alt+O 进入选择菜单

Alt+P 进入工程菜单

Alt+R 运行程序

Alt+X 退出 Turbo C 回到 DOS

Alt+F5 交换存储的屏幕

现在您已拥有源程序文件 exam1_1.c 了。注意，该文件在 C 盘上。如果您想将文件保存在 A 盘上，则在编辑时可修改盘号，如下所示：

```
C:\TC>tc A:exam1_1.c
```

1.3.2 在 UNIX 环境下

按照下列步骤建立源程序文件 exam1_1.c:

1.键入 vi exam1_1.c。注意，此处后缀.C 不能省略，这和 TURBO C 环境不同。

```
% vi exam1_1.c
```

其中%或\$是 UNIX 提示符。

2.键入 vi exam1_1.c 时，进入编辑状态。vi 编辑状态有两种工作方式：命令方式和输入方式。为了输入源程序，必须进入输入方式。键入 i(insert)或 a(append)进入输入方式(注意：键入的 i 或 a 并不在屏幕上显示)，程序输入完毕时，按 ESC 键，回到命令方式，然后按 ZZ(按大写字母 Z 两次)存盘并退出编辑状态。

如果发现程序有错，则可再次进入编辑状态。然后使用下列编辑命令对程序进行修改。

删除一个字符——将光标移到欲修改字符处，按 X 键。

删除 n 个字符——将光标移到欲修改字符处，按 nX 键，则从当前光标开始的右边 n 个字符被删除(n 为正整数)。

替代一个字符——光标移至欲替代字符处，按 r 键，然后键入新的字符。

替代 n 个字符——光标移至欲替代字符处，按 shift+r 然后键入 n 个新字符(n 为正整数)。必须按 ESC 键返回命令状态。

删除一行——将光标移到要删除的行，按 dd(即两次 d 键)。

删除 n 行——将光标移到要删除的开始行，按 n dd(n 为正整数)。

删除一行的一部分——将光标移到要删除的部分，按 shift+d，则自光标起至行末的部分被删除。

插入 n 个字符——将光标移到某字符处，按 i (在该字符前插入) 或 a (在该字符后插入)，插入结束后必须按 ESC 键返回命令方式。

插入一行——将光标移到某行，按 o(小写字母)，则在该行后出现一空行，您可输入字符，插入结束后必须按 ESC 返回命令方式。按 shift+o，则在该行前插入。

窗口移动——按 shift+f，窗口向下移动一屏，通常为 24 行。

按 shift+b，窗口向上移动一屏，

按 shift+d，窗口向下移动半屏，

按 shift+u，窗口向上移动半屏，

对于初学者，上述编辑命令已能应付编写程序。详细的编辑命令使用可参考 UNIX 操作系统使用手册。

如果您认为对程序的修改有效，则按 ZZ 存盘，修改有效并退出编辑。如果您想使本次修改无效，则按 shift+: 键，此时光标定位在屏幕底部，然后按 q! 退出编辑状态。此时原文件内容未作修改。

关于进入和退出 vi 编辑程序的命令归纳如下：

vi <文件名>——进入编辑状态，并处于命令方式。

ZZ——文件存盘，修改有效并退出编辑状态。

:wq<CR>——同 ZZ。<CR>表示按 Enter 键。

:w<CR>——文件存盘，不退出编辑状态。

:q!<CR>——文件不存盘，退出编辑状态。

§ 1.4 编译和运行 C 语言程序

1.4.1 编译和运行 C 语言程序的基本流程

C 语言程序是不能直接运行的，必须通过 C 编译程序对源程序进行编译，产生目标代码(又称目标程序)，目标代码是与源程序对应的机器可识别的二进制代码。目标代码存放在一个文件中，该文件称为目标文件。目标程序仍然不能执行，必须通过 C 连接程序把目标程序及该程序调用的库函数连接起来，构成可执行的目标代码(又称可执行程序)，然后才能执行。可执行的目标代码存放在一个文件中，该文件称为可执行文件。上述过程

可以用下面的流程图来示意:

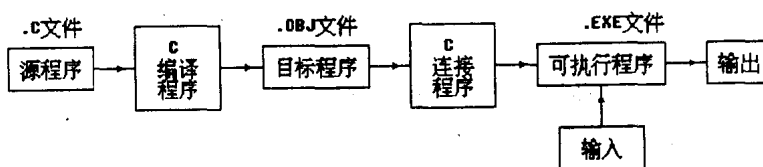


图 1.3 编译和运行 C 语言程序的流程图

1.4.2 在 DOS TURBO C 环境下编译和运行 C 程序

在 DOS TURBO C 环境下编译和运行 C 程序有两种方法: 菜单方式和交互方式。下面分别介绍这两种方式。

1.4.2.1 菜单方式

1. 编译

首先键入 tc <文件名> 进入主屏幕:

C:\TC>tc exam1_1

然后按 ALT+C 键, 对源程序进行编译, 此时屏幕中上部出现一个下拉窗口, 如下所示:

```
Compile to OBJ      C: EXAM1_1.OBJ
Make EXE file       C: EXAM1_1.EXE
Link EXE file
Build all
Primary C file
Get info
```

图 1.4 编译窗口

光标定位在第一行, 按 Enter 键开始编译。若无错误存在, 则在屏幕中下部出现一个窗口, 如下所示:

```
Compiling
Main file : C:EXAM1_1.C
Compiling : EDITOR -> EXAM1_1.C
Line Compiled :      4      4
Warnings      :      0      0
Errors        :      0      0
Available memory : 278K
Success       :      press any key
```

图 1.5 编译信息窗口

最后一行的 Success 表示编译成功, 此时可按任意键结束编译, 回到主屏幕。如果编

译有错，则在上述窗口的 Errors 行中会出现错误数量，最后一行不出现 Success，而是出现 Errors，按任意键后在屏幕的 Message 部分会提示那一行出错，什么性质的错误。此时需要重新进入编辑状态，修改源程序。

2. 连接和运行

如果编译成功，则按 ALT+R 键，对目标代码进行连接，并执行连接后的可执行代码。此时屏幕中上部出现一个下拉窗口，如下所示：

Run	Ctrl-F9
Program reset	Ctrl-F2
Go to cursor	F4
Trace into	F7
Step over	F8
User screen	ALT-F5

图 1.6 运行窗口

光标定位在窗口第一行上，按 Enter 键开始连接和运行。如果运行中无错误存在，则运行结束并返回主屏幕。您若想查看运行结果，可按 ALT+F5 键。看完结果后，只要按 Enter 键，又自动返回主屏幕。

如果运行中出现错误，则屏幕中部会出现一个窗口，如下所示：

Linking		
EXE file :	EXAM1_1.EXE	
Linking :	LIB\CS.LIB	
	Total	Link
Lines compiled :	0	pass2
Warning :	0	0
Errors :	1	1
Available memory:	215K	
Errors :	press any key	

图 1.7 连接信息窗口

按任意键后，可在屏幕 Message 部分看到错误的内容性质。遇到这种情况，需分析源程序，找出产生错误的原因，修改后重新进行编译和运行。

1.4.2.2 交互方式

1. 编译和连接

在交互方式下编译程序可简单地键入 tcc <文件名>，如下所示：

```
C:\TC>tcc exam1_1.c
```

如编译无错，则自动生成后缀为.OBJ 的目标文件 exam1_1.obj 和后缀为.exe 可执行 exam1_1.exe。您也可以更改目标文件名和可执行文件名。例示，若键入：

```
C:\TC>tcc -oabc -edef exam1_1.c
```

则生成 abc.obj 的目标文件和 edf.exe 的可执行文件。(注意-o 和 a 之间-e 和 d 之间均无空格)。

如果编译出错，则会在屏幕上显示出错信息。

2.运行

要运行程序只要简单地键入可执行文件名即可。

C:\TC>exam1_1.exe 或

C:\TC>exam1_1

§ 1.5 如何在屏幕上显示信息

在 § 1.2 中我们已经学会了编写在屏幕上显示字符串的 C 语言程序。在屏幕上显示字符串是通过调用 printf 函数实现的。这一节里,我们将通过改写 printf 语句,在屏幕上显示不同格式的字符串。如果想在不同的行上显示字符串,则有下列的 printf 语句:

```
printf("Hello \n");
```

```
printf("everybody !");
```

用这两个 printf 语句代替例 1.1 中的 printf 语句,则程序执行时,在屏幕上显示:

Hello

everybody !

为什么会有这样的结果呢?这是由于第 1 个 printf 语句中使用了转义字符 \n。凡是倒斜杠(\)后面跟上某个字母,例如字母 n,就构成了转义字符。转义字符都有特定的意义,转义字符 \n 表示换行。注意,在倒斜杠和字母之间不能出现空格。

如果去掉第 1 个 printf 语句中的 \n,并用空格代替,即

```
printf("Hello ");
```

```
printf("everybody !");
```

则结果和例 1.1 完全一样。可见,如果第 1 个 printf 语句中不出现 \n,则第 2 个 printf 语句中的字符串将与第 1 个 printf 语句中的字符串在同一行上输出。程序执行时,屏幕上将显示:

Hello everybody !

上面 printf 语句中的字符串是从屏幕上第 1 列开始显示的,如果想从第 8 列开始显示字符串,则可在字符串前面插入适当数量的空格。例如:

```
printf("□□□□□□□□Hello everybody !");
```

其中□表示空格。

另外一种方法是在字符串中插入转义字符 \t, \t 是制表字符,遇到 \t 时,光标将向右移动 8 列(移动列数由系统生成时设置),于是在当前字符的右边第 8 列上显示字符串的余下部分。例如:

```
printf("\t Hello everybody !");
```

将从第 8 列开始显示字符串:

Hello everybody!

如果 printf 语句改为:

```
printf("\t Hello \t everybody !");
```

则从第 8 列开始显示“Hello”,然后光标向右移动 8 列,再显示“everybody !”。如下所示:

Hello□□□□□□□everybody!

如果字符串太长，一行写不下，怎么办呢？可以使用转义字符‘\回车’，这里的回车表示按键盘上的 Enter 键或 Return 键。转义字符‘\回车’表示字符串尚未结束，下一行继续。例如：

```
printf("Hello everybody! Welcome to \
join the first East Asian Games!");
```

则在屏幕上显示：

Hello everybody! Welcome to join the first East Asian Games!

注意：printf 语句中第 1 行倒斜杠(\)后面有一个回车键。回车键本身在屏幕上是不显示的。按下回车键后，光标定位在下一行第 1 列上。

第一章 习题

(答案文件 exs1_1.c~exs1_10.c)

1. 试编制显示 C Language Programming 600 的程序。

2. 试编制显示：Test.....1.....2.....3 的程序。

3. 试编制输出如下所示信息的程序。

```
A B C D
    E F G
        H I J K L   M N O
            P Q R
                Z
S T U V   W X Y
```

4. 试编制输出如下所示信息的程序。

‘Mr. Shen’

‘Mr. Liu’

5. 试编制输出如下所示信息的程序。

A B C D E F G H I J K L

6. 试编制输出如下所示字符串的程序。

‘Schubert’, ‘Shumann’

‘Mozart’, ‘Beethovens’

‘Bach’, ‘Chopin’

7. 试编制输出如下所示字符串图案的程序。

```
/ / / / / t / / / / /
/ / / / t t t / / / /
/ / / t t t t / / /
/ / / t t t t / / /
/ / / t t t t / / /
/ / / / /
```

8. 试编制输出如下所示信息的程序。